

Monitorovanie softvérových projektov

JÁN VALAŠKA

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
valaska[zavináč]svobi[.]sk*

Abstrakt. Vývoj softvéru v tímoch, či už na komerčné, alebo akademické účely je komplikovaná úloha. Extrémne rýchly pokrok v technológiách a metódach vývoja softvéru spolu so vzrastajúcou zložitou vyvíjaných aplikácií predstavuje veľké výzvy pre softvérových inžinierov. Tieto výzvy sú navyše skomplikované tým, že softvérové procesy sú zamerané na človeka, čím ich môžeme charakterizovať aj ako veľmi variabilné a nepredvídateľné. Všetky tieto fakty ústia do množstva neskoro odovzdaných a predražených projektov. V tejto eseji sa budem snažiť zamerať na faktory a veličiny, ktoré by mal projektový manažér sledovať, spôsoby ako ich monitorovať a kroky, ktoré je vhodné podniknúť v prípade, že neide všetko podľa plánu.

Úvod

Niet pochýb, že vývoj softvéru je časovo, ale aj finančne veľmi náročnou úlohou. Softvérový manažér čelí viacerým výzvam - extrémne rýchlemu pokroku v technológiách a metódach vývoja softvéru, vzrastajúcej zložitosti vyvíjaných aplikácií a najmä tomu, že vzhľadom na svoju orientáciu na človeka sú tieto projekty veľmi variabilné a nepredvídateľné. Preto nie je neobvyklým javom, ak je projekt odovzdaný neskoro alebo sa predraží.

Podľa Johnsona [4] v roku 1995 americké firmy vynaložili približne 59 miliárd dolárov na predražené projekty a ďalších 81 miliárd stratili na zrušených projektoch [4]. Iný výskum, ktorého sa zúčastnilo 72 vyvíjaných informačných systémov v 23 amerických firmách ukázal, že priemerne 36% projektov prekročilo odhadnutý počet pracovných hodín a 22% z týchto projektov bolo odovzdaných neskoro – po plánovanom termíne [3]. Myslím si, že z týchto štatistík jasne vyplýva potreba kvalitnejšieho sledovania a kontrolovania vývoja softvérových projektov.

Prečo je manažovanie softvérových projektov také zložité? Prečo je z nich toľko neúspešných? Medzi hlavné príčiny patria [5]:

- Nehmatateľnosť - softvér je na rozdiel od hardvéru nehmatateľný. Neobsahuje viditeľné míľniky, podľa ktorých možno merať jeho progres a kvalitu.
- Zložitosť - súčasné aplikácie sú pre „bežného“ človeka ťažko pochopiteľné, čo spôsobuje nielen technické, ale aj manažérske problémy.
- Nestálosť požiadavok - požiadavky na softvér sa neustále menia. Keďže softvér je ľahšie meniteľnejší ako hardvér, zmena je jednoducho súčasťou softvérového vývoja.

Popri vyššie uvedených prekážkach sa vo vedeckých článkoch často spomínajú aj nasledujúce problémy [5]:

- nejasne definované ciele a nekvalitné špecifikácie softvéru
- absencia projektového plánu
- nereálne termíny a rozpočty.

V tejto eseji sa budem snažiť zamerať na faktory a veličiny, ktoré by mal manažér sledovať, aby bola jeho práca efektívna. Opíšem spôsoby monitorovania a kroky, ktoré je vhodné podniknúť v prípade, že sa vývoj softvéru neuberá podľa stanoveného plánu.

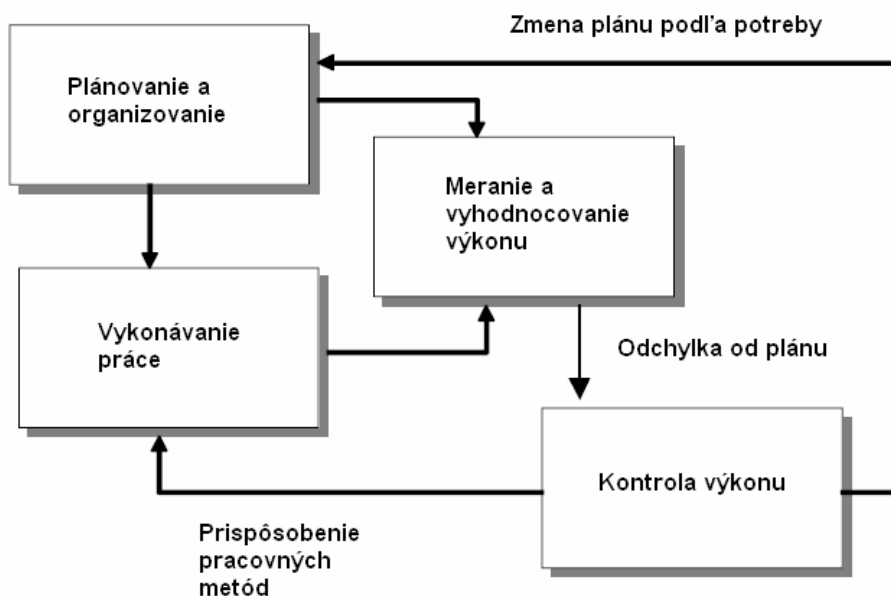
Monitorovanie a kontrola projektu

Cieľom monitorovania projektu je [5]:

- usmerňovať vývoj projektu tak, aby bol čo najbližšie k projektovému plánu
- identifikovať možné problémy ešte pred tým ako nastanú
- nasadiť potrebné opatrenia.

Základom monitorovania je porovnávanie vývoja s plánom a učiňovanie korekčných krokov v prípade odchýlky od plánu. Schéma monitorovania je zobrazená na obrázku č.1 [5].

Pre včasné a efektívne reagovanie na prípadné problémy musí mať softvérový manažér dokonalý prehľad o vývoji projektu, čo si vyžaduje jeho systematické monitorovanie. Hĺbka a frekvencia monitorovania závisí od typu a veľkosti projektu, pričom každý projekt sa dá monitorovať z rôznych pohľadov - aspektov. Medzi základné ukazovatele patria: postup, náklady, odchýlky od harmonogramu, funkcionality, chybovosť a stálosť požiadavok.



Obr. 1. Schéma monitorovania projektu.

V nasledujúcej časti eseje sa pokúsím stručne objasniť podstatu vyššie uvedených ukazovateľov, vysvetliť spôsoby ich merania a načrtnúť možné riešenie neželaných stavov.

Sledovanie progresu - postupu

Pod postupom sa myslí miera plnenia naplánovaných úloh. Slovičko progres nám môže asociovať množstvo vykonanej práce. Je pravda, že práca môže, ale aj nemusí viesť k priblíženiu sa k stanoveným cieľom. Bercun [1] hovorí, že prácu možno merať napríklad počtom:

- plánovania budúcich aktivít a úloh
- odpracovaných hodín
- napísaných správ
- odprednášaných prezentácií
- uskutočnených telefónnych hovorov
- naprogramovaných riadkov kódu
- odstránených chýb.

Žiadna z týchto ľahko merateľných veličín však nevie odmerať o koľko, a či vôbec sa projekt posunul k svojmu cieľu. Pri meraní postupu projektu je nutné sa hlbšie zamyslieť a posnažiť sa formulovať otázky, ktoré by manažérovi s veľkou presnosťou povedali v akom stave sa aktuálne projekt nachádza. Podľa Bercuna [1] je účinné klásť členom tímu nasledujúce otázky:

- O koľko bližšie k ukončeniu projektu nás posunula Vaša dnešná práca?
- Pracovali ste dostatočne kvalitne?
- Aké problémy ste vyriešili?
- Akým problémom ste zamedzili vyniknúť?
- Čo bráni Vášmu rýchlejšiemu napredovaniu?

Z doteraz nazbieraných skúseností viem, že veľmi vhodnou príležitosťou pre manažéra pre zisťovanie (nielen pokroku projektu) sú pravidelné projektové stretnutia. Vďaka nim sa dozvie o pokroku na jednotlivých častiach projektu, môže tak tiež odhaliť odchýlky od plánu a následne vykonať potrebné nápravné kroky. Tieto stretnutia sú príležitosťou na diskusiu o aktuálnych a prípadných budúcich problémoch a ich riešeníach a v neposlednom rade ponúkajú priestor pre komunikáciu medzi jednotlivými členmi tímu. Na každom stretnutí by sa mal zúčastniť aspoň jeden člen z každého tímu, ktorý je schopný odpovedať na prípadné otázky, navrhovať riešenia a diskutovať o problémoch.

Na stretnutiach sa jasne definujú ciele a plán projektu a pridelia sa úlohy každému členovi, ktoré by mal splniť do nasledujúceho stretnutia. Stretnutia sú najefektívnejšie, ak sa uskutočňujú pravidelne a vždy je dodržaný približne rovnaký scenár. Každé stretnutie by malo pozostávať zo:

- zistenia aktuálneho stavu najrizikovejších častí projektu
- zhodnotenia celkového progresu projektu
- zhodnotenia vývoja jednotlivých častí projektu
- plánovania budúcich aktivít a úloh.

Sledovanie časového plánu

Monitorovanie časového plánu predstavuje porovnávanie aktuálneho stavu projektu s projektovým plánom. Zaoberá sa ovplyvňovaním faktorov, ktoré spôsobujú zmenu časového plánu, zisťovaním a manažovaním zmien v prípade, že tieto nastanú. Vstupmi sú časový plán, výkonnostné správy a požiadavky na zmeny. Medzi nástroje a techniky používané pri monitorovaní časového plánu patria [2]:

- výkonnostné merania
- projektové stretnutia
- dodatočné plánovanie.

Na základe posúdenia výsledkov vyššie uvedeného sa dá určiť, či sme alebo nie sme v časovom sklze. V niektorých prípadoch sa dajú zistené sklzy v časovom harmonograme napraviť motivovaním zaangažovaného tímu, alebo pribratím ďalších pracovných síl pre danú úlohu. V prípade, že tento postup nieje možný a jedná o zmeny termínov odovzdaní častí projektu, je potrebné diskutovať zamýšľané zmeny s klientom. Z každého neskoro odovzdaného projektu by sa mal projektový manažér poučiť. Vhodné je viesť si záznamy popisujúce, z akých dôvodov sa nepodarilo dodržať stanovené termíny a snažiť sa týmto problémom v budúcnosti vystríhať.

Sledovanie nákladov

Monitorovanie nákladov sa zaoberá porovnávaním doteraz vynaložených nákladov na jednotlivé fázy s počiatočnými odhadmi. Základom je vykonávanie nasledujúcich aktivít [2]:

- monitorovanie vývoja nákladov, na zachytenie odchýlok od plánu
- ubezpečovanie sa, že všetky vykonané zmeny sú započítané v celkovom pláne
- informovanie klienta o zmene nákladov na produkt.

Náklady sa vo vývoji softvéru reprezentujú ako tzv. „človeko-hodiny“. Pri sledovaní nákladov by mal manažér vychádzať z odhadov, plánu nákladov, požadovaných zmien a výkonnostných správach. Podľa môjho názoru sú najväčším problémom precenené, alebo podcenené odhady hodín na jednotlivé úlohy. V prípade precenenia hrozí, že dodávateľ príde o projekt, pretože niekto iný môže byť schopný ponúknuť lepšiu kalkuláciu nákladov ako on. Pri podhodnotení musí dodávateľ hradiť rozdiel medzi odhadovanými a skutočnými nákladmi.

V súčasnosti sa používa *Technika zarobenej hodnoty* (Earned Value Method), ktorá je zlepšením tradičného porovnávania medzi predbežnou kalkuláciou a skutočnými nákladmi za účelom sledovania vývoja v projekte. EVM sa zameriava na vzájomný vzťah medzi vynaloženými skutočnými nákladmi, oproti skutočne realizovaným prácam v projekte, pre schopnosť získania ekonomickej hodnoty vykonanej práce, tzn. hodnoty prínosu. EVM vyžaduje jednotnú metódu pre analýzy a poskytuje podklady pre analýzu efektívnosti nákladov v projekte. Riadenie projektu tým získava lepšiu kontrolu a prehľad o vývoji projektu [6].

Sledovanie funkcionality

Sledovanie funkcionality je proces, kde o snahu ubezpečiť sa, že sú splnené všetky technické požiadavky na vyvíjaný softvér. Vstupmi pre tento proces sú výsledky práce a dokumentácia k projektu [2]. Úroveň dokončenia dôležitých funkcií sa meria pomocou najrôznejších metrík. Výsledky získané pomocou týchto meraní ponúkajú manažérovi predstavu o tom, čo bolo urobené. Na základe týchto výsledkov sa dá potom odhadnúť, čo ešte je možné urobiť v budúcnosti. V praxi sa veľmi často

používajú tzv. revízne prehliadky, ktoré pozostávajú z merania, skúšania a testovania projektu. Tieto prehliadky sa väčšinou vykonávajú po ukončení jednotlivých etáp vývoja. Z môjho pohľadu efektívnejšou technikou, ktorú možno použiť pri menších projektoch je zisťovanie stavu danej funkcie na projektových stretnutiach od zodpovedného člena tímu. Je potrebné však zväžiť schopnosť reálneho zhodnotenia stavu projektu zodpovednými osobami.

Výstupmi sledovania funkcionality je formálne osvedčenie projektu o plnení klientom požadovanej funkcionality [2].

Sledovanie kvality

Sledovanie kvality zabezpečuje, aby vytváraný produkt ponúkal nielen požadovanú funkcionality, ale aby bola aj v dostatočnej kvalite. Technické a kvalitatívne požiadavky by mali byť definované jasne a merateľne tak, aby im rozumel samotný klient a zároveň aj vývojový tím. Hlavným poslaním monitorovania kvality je detekcia a oprava chýb v čo najskoršom štádiu a udržiavanie čitateľného a ľahko rozšíriteľného kódu. Včasné nájdenie a odstránenie chýb výrazne znižuje problémy s nákladmi a harmonogramom.

Za vhodný spôsob vyjadrenia chybovosti považujem podiel počtu riadkov kódu a počet chýb. Vysoká chybovosť kódu môže byť spôsobená rôznymi príčinami. Medzi najčastejšie patrí práca pod tlakom a neskúsenosť. Jedným z veľmi efektívnych spôsobov znižovania chybovosti kódu je okrem samotného testovania aj kontrola vyprodukovaného kódu skúsenejšími kolegami.

Druhým meradlom kvality aplikácie je čitateľnosť a rozšíriteľnosť kódu. Programátori majú vo zvyku používať rôzne skrátene zápisy, ktoré síce vyzerajú sofistikovanejšie, ale ich čitateľnosť je pre iných veľmi nízka. Pri neokomentovaní kódu už ani samotný autor o pár týždňov zväčša nevie načo daný kus kódu slúži a začínajúci programátor v tíme už vôbec nie. Stáva sa, že kód je v tak zlom stave, že sa prerušia všetky práce na vývoji funkcionality a je nutné celý kód sformátovať, refaktorovať, optimalizovať a vhodne okomentovať.

Monitorovanie stálosti požiadaviek

Aj tú najdokonalejšiu špecifikáciu bude treba doplniť, či zmeniť po návrhu a testovaní produktu. Z vlastnej skúsenosti viem, že zmena funkcionality, respektíve „došpecifikovanie“ produktu po počiatočnej analýze a schválení odhadu robí vrásky na čele nie jednému softvérovému manažérovi, či developerovi. Pri nekontrolovanom pozmeňovaní špecifikácie je takmer isté, že sa výsledný projekt nedodrží vytvorený harmonogram, alebo prekročí stanovený rozpočet. Prečo hovorím o tom, že náklady sa zvýšia a neberiem do úvahy možnosť, že by sa mohli aj znížiť? Ešte som sa nestretol s tým, že by zákazník redukoval funkcionality po špecifikovaní projektu.

Osobne si myslím, že vhodným riešením je tvrdý prístup k zákazníkovi. To čo bolo raz špecifikované, analyzované a odhadnuté by sa už nemalo meniť. Dodatočne

požadované zmeny si môže zákazník samozrejme objednať avšak ako novú zákazku (tzv. „enhancement“).

Častým problémom takisto býva nekvalitná špecifikácia na základe ktorej je vytvorený počiatočný odhad hodín. Počas analýzy sa však zistí, že sa pri odhade nerátalo s mnohými skrytými funkciami.

Vyhodnocovanie projektu

Vyhodnocovanie projektu je periodické posudzovanie stavu projektu vzhľadom na požadované ciele. Tento proces sa od monitorovania projektu líši v tom, že za monitorovanie je zodpovedný projektový manažér a za vyhodnocovanie nadradený manažment. Líšia sa aj v tom, že vyhodnocovanie sa uskutočňuje iba raz za určité obdobie, zatiaľ čo monitorovanie projektu je neustály proces. Obvykle sa vyhodnocovanie projektu uskutoční pri tzv. „release“ verzii, alebo odovzdaní celého projektu, preto, aby sme zistili, či nie je potrebné vykonať určité dôležité zmeny v projekte. Pod týmito zmenami rozumiem prehodnotenie cieľov projektu, prerobenie plánu, alebo zrušenie, či pozastavenie projektu [5].

Záver

Monitorovanie je jednou z kľúčových aktivít manažéra a to nielen pri vývoji softvérových produktov. V tejto eseji som uviedol rôzne aspekty a techniky, ktoré sú používané pre kontrolu vývoja softvéru. Pre optimálne dodržanie stanoveného časového a finančného plánu je potrebné, aby projektový manažér dohliadal na neustálu kontrolu jednotlivých spomenutých aspektov. Svoje rozhodnutia by mal prispôbovať aktuálnemu stavu projektu podľa konkrétnych zistení. Nie je možné nájsť optimálny model pre monitorovanie projektov, ale pri dodržaní uvedených zásad je pravdepodobnejšie, že po odovzdaní projektu budú spokojné obe strany.

Použitá literatúra

1. Bercun, Scott: Work vs. Progress, August 2005
<http://www.scottbercun.com/essays/essay45.html>
aktuálne k 24.10.2007
2. Duncan, William R.: *A Guide to the Project Management Body of Knowledge*, Project Management Institute, 1996.
3. Genuchten, M.: *Why is Software Late? An Empirical Study of Reasons for Delay in Software Development*, IEEE Transactions on Software Engineering, Vol. 17, No. 6, June 1991, p. 582-590.
4. Johnson, J.: *Chaos: The Dollar Drain of IT Project Failures*, Application Development Trends, Vol. 2, No.1, 1995, p. 41-47.
5. Jurison, J.: *Software Project Management: The Manager's View*. Communications of the Association for Information Systems, Vol. 2, Article 17, September 1999.

6. Kováčiková, M.: *Earned Value Management*, 2006.
<http://www.fpedas.utc.sk/katedry/ks/casopis/pdf/IV2006/kovacikova.pdf>
aktuálne k 24.10.2007
7. University of southern California, Software Engineering II, 2001.
http://sunset.usc.edu/classes/cs577b_2001/metricsguide/metrics.html#p33
aktuálne k 24.10.2007

Annotation

Monitoring software projects

Software development in teams for business or academic purposes is complex task. Extremely rapid evolution in technologies and methods for software development together with increasing complexity of developed applications introduce new challenges for software engineers. These challenges are even more complicated by the fact that software processes are human oriented, due to which they can be characterized as very unstable and unpredictable. All these facts contribute to a huge number of delayed or over budgeted projects. In this paper we would try to focus on factors and variables, which should be considered by project manager, techniques of measuring them and appropriate actions, which should be applied in case that something goes wrong.